Claims

1. Method for optimizing the order of assignment of a number of supplies or resources, such as computer processor units, to a number of demanders or demands, such as tasks to be processed by the computer processor units, where each supply or resource has a certain supply or resource amount, such as a processing capacity, and each demander has a certain demand amount which is to be satisfied by said supplies or resources, such as a capacity demand, i.e. the processing capacity necessary to process the task, the method comprising the steps of:
   - building a network,
     in which the supplies are represented by supply vertices connected to a sink vertex via sink edges of a flow capacity which represents the supply amount of the respective supply, the sink vertex being sink of a network flow;
     in which the demanders are represented by demander vertices connected to a source vertex via source edges of a flow capacity which represents the demand amount of the respective demander, the source vertex being source of a network flow; and
     in which supply vertices and demander vertices are connected by edges of certain flow capacities;
   - determining an optimized network flow distribution of flow values through the edges by an iterative flow-method; and
   - deriving the optimized order of assignment from the optimized network flow distribution by assigning the supply vertices to the demander vertices in correspondence to the flow values of the connecting edges.

2. Method according to claim 1, wherein in the iterative flow-method comprises a discharge operation pushing a flow from an active vertex at which the sum of the incoming network flow is higher than the sum of the outgoing network flow along an admissible edge, where the admissibility of an edge is defined by a label of the vertex connected to the active vertex by the respective edge.

3. Method according to claim 2, further comprising a relabeling operation changing the label of the active vertex if there is no admissible edge along which the discharge operation can be performed.

4. Method according to claim 3, wherein, when the label of the vertex to be discharged is $\Psi(v)$ and the label of a vertex connected by an edge is $\Psi(w)$, said edge being admissible if $\Psi(v) = \Psi(w) + 1$, and wherein the label $\Psi(v)$ of the vertex to be discharged is increased by one in the relabeling operation.

5. Method according to any of the claims 2 to 4, comprising discharge operations pushing flows from demander vertices to supply vertices and discharge operations pushing flows from supply vertices to demander vertices.

6. Method according to claim 5, wherein the discharge operation is performed iteratively for demander vertices and supply vertices.

7. Method according to any of the preceding claims, wherein a total flow is the network flow through the edges from the source vertex to the sink vertex and the flow-method comprises the steps of:
   - determining an upper limit of the highest possible total flow through the edges; and
   - iteratively distributing the network flow through the edges until at least one of the conditions is fulfilled:
     i) the network flow corresponds to the upper limit of the highest possible total flow,
     ii) the sum of the incoming network flow at a vertex equals the sum of the outgoing network flow of said vertex for each supply vertex and for each demander vertex,
     iii) the number of iterations has reached a given maximum value.

8. Method according to any of the preceding claims, wherein assigning the supply vertices to the demander vertices is performed by an iterative assigning operation.

9. Method according to claim 8, wherein the assigning operation, in a first stage, assigns a supply vertex to a demander vertex only if these vertices are connected by an edge for which the flow value equals the capacity.

10. Method according to claim 9, wherein the assigning operation first assigns supply vertices to such demander vertices which are connected to the respective supply vertex by an edge for which the flow value equals the flow value of the corresponding source edge before it assigns supply vertices to such demander vertices which are connected to the respective supply vertex by an edge for which the flow value is equal to or higher than a remaining flow value of the corresponding sink edge which has not yet been assigned to a demander vertex.

11. Method according to any of the claims 9 and 10, wherein the first stage is performed until all supply vertices and demander vertices which are connected by edges for which the flow value equals the capacity are assigned.

12. Method according to claim 11, wherein the assigning operation, in a second stage, assigns a supply vertex to a demander vertex if the flow value of the connecting edge corresponds to the flow value of the corresponding source edge reduced by a fraction of its demand amount already assigned to a supply vertex, or to the flow value of the corresponding sink edge reduced by a fraction of its supply amount already assigned to a demander vertex.

13. Method according to claim 12, wherein the assigning operation, in the second stage, first assigns such supply vertices to demander vertices for which the flow value of the connecting edge corresponds to the flow value of the corresponding source edge reduced by a fraction of its demand amount which is already assigned to a supply vertex.

14. Method according to any of the preceding claims, wherein the certain flow capacity of an edge connecting a demander vertex to a supply vertex is given by the smaller one of the capacity of the respective source edge and the capacity of the respective sink edge.

15. Method for balancing a number of loan accounts with a number of collateral securities, where each loan account has a certain loan value and each collateral security has a certain security value and wherein the collateral securities are to be offset against the loan accounts, the method comprising the steps of:
    - building a network
      in which the collateral securities are represented by security vertices connected to a sink vertex via sink edges of a flow capacity which represents the security value of the respective collateral security, the sink vertex being sink of a network flow;
      in which the loan accounts are represented by loan vertices connected to a source vertex via source edges of a flow capacity which represents the loan value of the respective loan account, the source vertex being source of a network flow; and
      in which security vertices and loan vertices are connected by edges of certain flow capacities;
    - determining an optimized network flow distribution of flow values through the edges by an iterative flow-method; and
    - deriving the optimized order of assignment from the optimized network flow distribution by offsetting the security vertices against the loan vertices in correspondence to the flow values of the connecting edges.

16. Method according to claim 15, wherein the iterative flow-method comprises a discharge operation pushing a flow from an active vertex, at which the sum of the incoming network flow is higher than the sum of the outgoing network flow, along an admissible edge, where the admissibility of an edge is defined by a label of the vertex connected to the active vertex by the respective edge.

17. Method according to claim 16, further comprising a relabeling operation changing the label of the active vertex if there is no admissible edge along which the discharge operation can be performed.

18. Method according to claim 17, wherein, when the label of the vertex to be discharged is $\Psi(v)$ and the label of a vertex connected by an edge is $\Psi(w)$, said

edge being admissible if $\Psi(v) = \Psi(w) + 1$, and wherein the label $\Psi(v)$ of the vertex to be discharged is increased by one in the relabeling operation.

19. Method according to any of the claims 16 to 18, comprising discharge operations pushing flows from loan vertices to security vertices and discharge operations pushing flows from security vertices to loan vertices.

20. Method according to claim 19, wherein the discharge operation is performed iteratively for loan vertices and security vertices.

21. Method according to any of the claims 15 to 20, wherein a total flow is the network flow through the edges from the source vertex to the sink vertex and wherein the flow-method comprises the steps of:
   - determining an upper limit of the highest possible total flow through the edges; and
   - iteratively distributing the network flow through the edges until at least one of the conditions is fulfilled:
     i) the network flow corresponds to the upper limit of the highest possible total flow,
     ii) the sum of the incoming network flow at a vertex equals the sum of the outgoing network flow of said vertex for each security vertex and for each loan vertex,
     iii) the number of iterations has reached a given maximum value.

22. Method according to any of the claims 15 to 21, wherein offsetting the security vertices against the loan vertices is performed by an iterative offsetting operation.

23. Method according to claim 22, wherein the offsetting operation in a first stage offsets a security vertex against a loan vertex only if these vertices are connected by an edge for which the flow value equals the capacity.

24. Method according to claim 23, wherein the offsetting operation first offsets security vertices against such loan vertices which are connected to the

respective security vertex by an edge for which the flow value equals the flow value of the corresponding source edge before it offsets security vertices against such loan vertices which are connected to the respective security vertex by an edge for which the flow value is equal to or higher than a remaining flow value of the corresponding sink edge which has not yet been offset against to a loan vertex.

25. Method according to any of the claims 23 and 24, wherein the first stage is performed until all security vertices and loan vertices which are connected by edges for which the flow value equals the capacity are offset.

26. Method according to claim 25, wherein the offsetting operation, in a second stage, offsets a security vertex against al loan vertex if the flow value of the edge connecting the respective vertices corresponds to the flow value of the corresponding source edge reduced by a fraction of its loan value already offset against a security vertex, or to the flow value of the corresponding sink edge reduced by a fraction of its security value already offset against a loan vertex.

27. Method according to claim 26, wherein the offsetting operation, in the second stage, first offsets such security vertices against loan vertices for which the flow value of the edge connecting the respective vertices corresponds to the flow value of the corresponding source edge reduced by the loan value already offset against a collateral security.

28. Method according to any of the claims 15 to 27, wherein the certain flow capacity of an edge connecting a loan vertex to a sink vertex is given by the smaller one of the capacity of the respective source edge and the capacity of the respective sink edge.

29. Method for optimizing the data transfer through a transmission system comprising a number of senders and a number of transmission lines, where each transmission line has a certain transmission rate and each sender is connected to a number of said transmission lines and has a certain data rate, by assigning transmission lines to senders, the method comprising the steps of:

- building a network

  in which the transmission lines are represented by transmission vertices connected to a sink vertex via sink edges of a flow capacity which represents the transmission rate of the respective transmission line, the sink vertex being sink of a network flow;

  in which the senders are represented by sender vertices connected to a source vertex via source edges of a flow capacity which represents the data rate of the respective sender, the source vertex being source of a network flow; and

  in which transmission vertices and sender vertices are connected by edges of certain flow capacities;
- determining an optimized network flow distribution of flow values through the edges by an iterative flow-method; and
- deriving the optimized order of assignment from the optimized network flow distribution by assigning the transmission vertices to the sender vertices in correspondence to the flow values of the connecting edges.

30. Method according to claim 29, wherein in the iterative flow-method comprises a discharge operation pushing a flow from an active vertex at which the sum of the incoming network flow is higher than the sum of the outgoing network flow along an admissible edge, where the admissibility of an edge is defined by a label of the vertex connected to the active vertex by the respective edge.

31. Method according to claim 30, further comprising a relabeling operation changing the label of the active vertex if there is no admissible edge along which the discharge operation can be performed.

32. Method according to claim 31, wherein, when the label of the vertex to be discharged is $\Psi(v)$ and the label of a vertex connected by an edge is $\Psi(w)$, said edge being admissible if $\Psi(v) = \Psi(w) + 1$, and wherein the label $\Psi(v)$ of the vertex to be discharged is increased by one in the relabeling operation.

33. Method according to any of the claims 30 to 32, comprising discharge operations pushing flows from sender vertices to transmission vertices and

discharge operations pushing flows from transmission vertices to sender vertices.

34. Method according to claim 33, wherein the discharge operation is performed iteratively for sender vertices and transmission vertices.

35. Method according to any of the claims 29 to 34, wherein a total flow is the network flow through the edges from the source vertex to the sink vertex and the flow-method comprises the steps of:
- determining an upper limit of the highest possible total flow through the edges; and
- iteratively distributing the network flow through the edges until at least one of the conditions is fulfilled:
  i) the network flow corresponds to the upper limit of the highest possible total flow,
  ii) the sum of the incoming network flow at a vertex equals the sum of the outgoing network flow of said vertex for each transmission vertex and for each sender vertex,
  iii) the number of iterations has reached a given maximum value.

36. Method according to any of the claims 29 to 35, wherein assigning the transmission vertices to the sender vertices is performed by an iterative assigning operation.

37. Method according to claim 36, wherein the assigning operation, in a first stage, assigns a transmission vertex to a sender vertex only if these vertices are connected by an edge for which the flow value equals the capacity.

38. Method according to claim 37, wherein the assigning operation first assigns transmission vertices to such sender vertices which are connected to the respective transmission vertex by an edge for which the flow value equals the flow value of the corresponding source edge before it assigns transmission vertices to such sender vertices which are connected to the respective supply vertex by an edge for which the flow value is equal to or higher than a remaining

flow value of the corresponding sink edge which has not yet been assigned to a sender vertex.

39. Method according to any of the claims 37 and 38, wherein the first stage is performed until all transmission vertices and sender vertices which are connected by edges for which the flow value equals the capacity are assigned.

40. Method according to claim 39, wherein the assigning operation, in a second stage, assigns a transmission vertex to a sender vertex if the flow value of the connecting edge corresponds to the flow value of the corresponding source edge reduced by a fraction of its data rate already assigned to a transmission vertex, or to the flow value of the corresponding sink edge reduced by a fraction of its transmission rate already assigned to a sender vertex.

41. Method according to claim 40, wherein the assigning operation, in the second stage, first assigns such transmission vertices to sender vertices for which the flow value of the connecting edge corresponds to the flow value of the corresponding source edge reduced by a fraction of its data rate already assigned to a transmission vertex.

42. Method according to any of the claims 29 to 41, wherein the certain flow capacity of an edge connecting a sender vertex to a transmission vertex is given by the smaller one of the capacity of the respective source edge and the capacity of the respective sink edge.

43. Method for optimizing the order of assignment of a number of tasks to a number of processors, where each processor has a certain processor capacity and each task has a certain capacity demand which is to be satisfied by at least one of said processors, the method comprising the steps of:

- building a network
  in which the processors are represented by processor vertices connected to a sink vertex via sink edges of a flow capacity which represents the processor capacity of the respective processor, the sink vertex being sink of a network flow;

in which the tasks are represented by task vertices connected to a source vertex via source edges of a flow capacity which represents the capacity demand of the respective task, the source vertex being source of a network flow; and

in which processor vertices and task vertices are connected by edges of certain flow capacities;

- determining an optimized network flow distribution of flow values through the edges by an iterative flow-method; and
- deriving the optimized order of assignment from the optimized network flow distribution by assigning the processor vertices to the task vertices in correspondence to the flow values of the connecting edges.

44. Method according to claim 43, wherein in the iterative flow-method comprises a discharge operation pushing a flow from an active vertex at which the sum of the incoming network flow is higher than the sum of the outgoing network flow along an admissible edge, where the admissibility of an edge is defined by a label of the vertex connected to the active vertex by the respective edge.

45. Method according to claim 44, further comprising a relabeling operation changing the label of the active vertex if there is no admissible edge along which the discharge operation can be performed.

46. Method according to or claim 45, wherein, when the label of the vertex to be discharged is $\Psi(v)$ and the label of a vertex connected by an edge is $\Psi(w)$, said edge being admissible if $\Psi(v) = \Psi(w) + 1$, and wherein the label $\Psi(v)$ of the vertex to be discharged is increased by one in the relabeling operation.

47. Method according to any of the claims 44 to 46, comprising discharge operations pushing flows from task vertices to processor vertices and discharge operations pushing flows from processor vertices to task vertices.

48. Method according to claim 47, wherein the discharge operation is performed iteratively for task vertices and processor vertices.

49. Method according to any of the claims 43 to 48, wherein a total flow is the network flow through the edges from the source vertex to the sink vertex and the flow-method comprises the steps of:
    - determining an upper limit of the highest possible total flow through the edges; and
    - iteratively distributing the network flow through the edges until at least one of the conditions is fulfilled:
        i) the network flow corresponds to the upper limit of the highest possible total flow,
        ii) the sum of the incoming network flow at a vertex equals the sum of the outgoing network flow of said vertex for each processor vertex and for each task vertex,
        iii) the number of iterations has reached a given maximum value.

50. Method according to any of the claims 43 to 49, wherein assigning the processor vertices to the task vertices is performed by an iterative assigning operation.

51. Method according to claim 50, wherein the assigning operation, in a first stage, assigns a processor vertex to a task vertex only if these vertices are connected by an edge for which the flow value equals the capacity.

52. Method according to claim 51, wherein the assigning operation first assigns processor vertices to such task vertices which are connected to the respective processor vertex by an edge for which the flow value equals the flow value of the corresponding source edge before it assigns processor vertices to such task vertices which are connected to the respective processor vertex by an edge for which the flow value is equal to or higher than a remaining flow value of the corresponding sink edge which has not yet been assigned to a task vertex.

53. Method according to any of the claims 51 and 52, wherein the first stage is performed until all processor vertices and task vertices which are connected by edges for which the flow value equals the capacity are assigned.

54. Method according to claim 53, wherein the assigning operation, in a second stage, assigns a processor vertex to a task vertex if the flow value of the connecting edge corresponds to the flow value of the corresponding source edge reduced by a fraction of its capacity demand already assigned to a processor vertex, or to the flow value of the corresponding sink edge reduced by a fraction of its processing capacity already assigned to a task vertex.

55. Method according to claim 54, wherein the assigning operation in the second stage first assigns such processor vertices to task vertices for which the flow value of the connecting edge corresponds to the flow value of the corresponding source edge reduced by a fraction of its capacity demand which is already assigned to a processor vertex.

56. Method according to any of the claims 43 to 55, wherein the certain flow capacity of an edge connecting a task vertex to a processor vertex is given by the smaller one of the capacity of the respective source edge and the capacity of the respective sink edge.

57. Device for determining an optimized assignment of a number of supplies or resources, such as computer processor units, each having a certain supply or resource amount, such as a processing capacity, to a number of demanders or demands, such as tasks to be processed by the computer processor units, each having a certain demand amount to be satisfied by said supplies, such as a capacity demand, i.e. a processing capacity necessary to process the task, in which, after the assignment, the sum of unsatisfied demand amounts is minimized, comprising:

- a supply input unit for inputting supply data representing supplies and their supply amounts;
- a demander input unit for inputting demander data representing demanders and their demand amounts,
- an access input unit for inputting access data representing, for each demander, the corresponding supplies which can be accessed by the respective demander for satisfying its demand amount;

- a network construction unit for constructing, on the basis of the supply data, the demander data and the access data, a network comprising:
  a) a supply vertex for each supply,
  b) a demander vertex for each demander,
  c) a sink vertex,
  d) a source vertex,
  e) edges, each having a certain flow capacity and connecting a supply vertex and a demander vertex,
  f) sink edges, each connecting the sink vertex to one of the supply vertices and having a flow capacity representing the supply amount of the respective supply, and
  g) source edges, each connecting the source vertex to one of the demander vertices and having a flow capacity representing the demand amount of the respective demander;
- a network flow unit for determining an optimized network flow distribution through the network, the optimized network flow being represented by flow values through the edges; and
- an assignment unit for assigning the supplies to the demanders by assigning the supply vertices to the demander vertices in correspondence to the flow values of the connecting edges.

58. Device according to claim 57, wherein the input units are formed by a single input unit.

59. Device according to claim 57, wherein the input units are integrated into a single device.

60. Device according to any of the claims 57 to 59, wherein the network construction unit, the network flow unit, and the assignment unit are realized by a single calculator unit.

61. Device for balancing a number of loan accounts with a number of collateral securities, where each loan account has a certain loan value and each collateral

security has a certain security value and wherein the collateral securities are to be offset against the loan accounts, comprising:

- a security input unit for inputting security data representing collateral securities and their security values;
- a loan input unit for inputting loan data representing loan accounts and their loan values,
- an access input unit for inputting access data representing, for each loan account, the corresponding collateral securities, which can be offset against the respective loan account;
- a network construction unit for constructing, on the basis of the security data, the loan data and the access data, a network comprising:
  a) a security vertex for each security,
  b) a loan vertex for each loan,
  c) a sink vertex,
  d) a source vertex,
  e) edges, each having a certain flow capacity and connecting a security vertex and a loan vertex,
  f) sink edges, each connecting the sink vertex to one of the security vertices and having a flow capacity representing the security value of the respective collateral security, and
  g) source edges, each connecting the source vertex to one of the loan vertices and having a flow capacity representing the loan value of the respective loan account;
- a network flow unit for determining an optimized network flow distribution through the network, the optimized network flow being represented by flow values through the edges; and
- an assignment unit for offsetting the collateral securities against the loan accounts by assigning the security vertices to the loan vertices in correspondence to the flow values of the connecting edges.

62. Device according to claim 61, wherein the input units are formed by a single input unit.

63. Device according to claim 61, wherein the input units are integrated into a single device.

64. Device according to any of the claims 61 to 63, wherein the network construction unit, the network flow unit, and the assignment unit are realized by a single calculator unit.

65. Device for determining an optimized data transfer through a transmission system comprising a number of senders and a number of transmission lines by assigning transmission lines to senders, where each transmission line has a certain transmission rate and each sender is connected to a number of said transmission lines and has a certain data rate, comprising:
   - a transmission line input unit for inputting transmission line data representing transmission lines and their transmission rates
   - a sender input unit for inputting sender data representing senders and their data rates
   - an access input unit for inputting access data representing, for each sender, the corresponding transmission lines, which can be accessed by a sender for transferring data;
   - a network construction unit for constructing, on the basis of the transmission line data, the sender data and the access data, a network comprising:
     a) a transmission vertex for each transmission line,
     b) a sender vertex for each sender,
     c) a sink vertex,
     d) a source vertex,
     e) edges, each having a certain flow capacity and connecting a transmission vertex and a sender vertex,
     f) sink edges, each connecting the sink vertex to one of the transmission vertices and having a flow capacity representing the transmission rate of the respective transmission line, and
     g) source edges, each connecting the source vertex to one of the sender vertices and having a flow capacity representing the data rate of the respective sender;

- a network flow unit for determining an optimized network flow distribution through the network, the optimized network flow being represented by flow values through the edges; and
- an assignment unit for assigning the transmission lines to the senders by assigning the transmission vertices to the sender vertices in correspondence to the flow values of the connecting edges.

66. Device according to claim 65, wherein the input units are formed by a single input unit.

67. Device according to claim 65, wherein the input units are integrated into a single device.

68. Device according to any of the claims 65 to 67, wherein the network construction unit, the network flow unit, and the assignment unit are realized by a single calculator unit.

69. Device for determining an optimized order of assignment of a number of tasks to a number of processors, where each processor has a certain processor capacity and each task has a certain capacity demand which is to be satisfied by at least one of said processors, comprising:
    - a processor input unit for inputting processor data representing processors and their processing capacities;
    - a task input unit for inputting task data representing tasks and their processing demands,
    - an access input unit for inputting access data representing, for each task, the corresponding processors, which can be accessed by a task;
    - a network construction unit for constructing, on the basis of the processor data, the task data and the access data, a network comprising:
        a) a processor vertex for each processor,
        b) a task vertex for each task,
        c) a sink vertex,
        d) a source vertex,

e) edges, each having a certain flow capacity and connecting a processor vertex and a task vertex,

f) sink edges, each connecting the sink vertex to one of the processor vertices and having a flow capacity representing the processing capacity of the respective processor, and

g) source edges, each connecting the source vertex to one of the task vertices and having a flow capacity representing the processing demand of the respective task;

- a network flow unit for determining an optimized network flow distribution through the network, the optimized network flow being represented by flow values through the edges; and
- an assignment unit for assigning the processors to the tasks by assigning the processor vertices to the task vertices in correspondence to the flow values of the connecting edges.

70. Device according to claim 69, wherein the input units are formed by a single input unit.

71. Device according to claim 69 wherein the input units are integrated into a single device.

72. Device according to any of the claims 69 to 71, wherein the network construction unit, the network flow unit, and the assignment unit are realized by a single calculator unit.

73. Computer program product for optimizing the order of assignment of a number of supplies to a number of demanders comprising instructions which, when loaded into a computer, cause said computer to perform a method as claimed in one of the claims 1 – 14.

74. Computer program product for balancing a number of loan accounts with a number of collateral securities comprising instructions which, when loaded into a computer, cause said computer to perform a method as claimed in one of the claims 15 – 28.

75. Computer program product for optimizing the data transfer through a transmission system comprising a number of senders and a number of transmission lines comprising instructions which, when loaded into a computer, cause said computer to perform a method as claimed in one of the claims 29 – 42.

76. Computer program product for optimizing the order of assignment of a number of tasks to a number of processors comprising instructions which, when loaded into a computer, cause said computer to perform the method as claimed in one of the claims 43 – 56.

77. Storage medium comprising stored data which represent a computer program product as claimed in claim 73.

78. Storage medium comprising stored data which represent a computer program product as claimed in claim 74.

79. Storage medium comprising stored data which represent a computer program product as claimed in claim 75.

80. Storage medium comprising stored data which represent a computer program product as claimed in claim 76.